International Journal of Mathematical Archive-2(5), May - 2011, Page: 802-806
Available online through www.ijma.info ISSN 2229 – 5046

# COMPUTERIZABLE STATISTICALLY-ORIENTED REDUCED-BIAS-SANDWICH REDUCED-MMRMSE ALGORITHMICALLY IMPROVED INTERPOLATION USING NEWTON'S FORWARD DIFFERENCE FORMULA

**[1]Winston A. Richards, *[2]Ashok Sahai & [3]Neeraj Tiwari**

*[1]Department of Mathematics and Statistics The Pennsylvania State University, Harrisburg, U. S. A.*
*E-mail: ugu@psu.edu*

*[2]Department of Mathematics & Statistics; Faculty of Science & Agriculture St. Augustine Campus; The University of The West Indies; Trinidad & Tobago.*
*E-mail: sahai.ashok@gmail.com*

*[3]Dept. of Statistics, Kumaun University, S.S.J.Campus, Almora, UTTARAKHAND - 01, INDIA.*
*E-mail: kumarn_amo@yahoo.com*

***(Received on: 03-05-11; Accepted on: 14-05-11)***

---

***ABSTRACT***

*This paper aims at proposing a sandwich-iterative computerizable numerical algorithm for an improved 'Interpolation Using Newton's Forward Difference Formula'. Without any loss of generality we might assume that the 'Interpolation-Interval' is C [0, 1]. The algorithm uses the 'statistical perspectives' for exploiting the information about the unknown function 'f', available in terms of its known values at the 'equidistant-knots in C [0, 1]', rather more fully. The improvement, achieved by an a-posteriori use of this information, gets to be happening iteratively. Any typical iteration uses the 'Twin Statistical Perspectives' of 'Relative Mean Square Error (RMSE)', & 'Bias'; the former concept being sandwiched by the latter. The potential of the achievable improvements through the proposed 'computerizable numerical iterative algorithm' is tried to be brought out/ illustrated per an 'empirical study' for which the function 'f' is assumed to be known in the sense of simulation.*

***Keywords & Phrases:*** *Bias, Relative mean square error, Simulated empirical study.*

***AMS Classification Numbers:*** *32E30, 41A05, 62F10.*

---

## 1. INTRODUCTION:

It is widely known that the 'Newton's Forward Difference Interpolation Formula' is very popularly used in the numerical analyses of many investigative case-studies in almost all fields of science & engineering.

This paper aims at proposing a sandwich-iterative computerizable numerical algorithm for an improved 'Interpolation Using Newton's Forward Difference Formula'. Without any loss of generality we might assume that the 'Interpolation-Interval' is C [0, 1] rather than any general interval, say C [a, b]. In fact, C [0, 1] and C [a, b] are essentially identical, for all practical purposes, inasmuch as they are linearly 'isometric' as normed spaces, order isomorphic as lattices, and isomorphic as algebras (rings)!

The proposed iterative-algorithm uses the 'Statistical Perspectives' for using the information about the unknown function 'f' available in terms of its known values at the 'equidistant-knots in C [0, 1]' rather more fully. The improvement, achieved by an a-posteriori use of this information, happens to be happening iteratively. Any typical iteration uses the 'Twin Statistical Perspectives' of 'Relative Mean Square Error (RMSE)', and 'Bias'; the former concept being sandwiched by the latter. Thus, at any typical 'Iteration' we firstly use the perspective of 'Reduced Bias', then secondly, we use the perspective of 'Reduced Relative Mean Squared Error (RRMSE)', and thirdly-n-lastly we use again the perspective of 'Reduced Bias'.

---

****Corresponding author: * Ashok Sahai, *E-mail: sahai.ashok@gmail.com***
*[2]Department of Mathematics & Statistics; Faculty of Science & Agriculture St. Augustine Campus; The University of The West Indies; Trinidad & Tobago.*

The potential of the achievable improvements through the proposed 'computerizable numerical iterative algorithm' is tried to be brought out/ illustrated per an 'empirical study' for which the function 'f' is assumed to be known in the sense of simulation.

**2. ITERATIVE-SANDWICH ALGORITHM FOR 'NEWTON'S FORWARD INTERPOLATION FORMULA':**

The algorithm's motivating 'Key-Point' is 'fuller-use' of the information about the unknown function 'f' available in terms of its known values at the 'equidistant-knots' in C [0, 1], namely "f(0/n), f(1/n), f(2/n), …, f(k/n), …, f(n/n)".

Newton's Forward Difference Interpolation Formula is having the following finite form, using the aforesaid values at the 'n+1 equidistant-knots' in the Interpolation-Interval C [0, 1]**.**

$f(x) = f(0) + {}^{x}C_1\, \Delta f(0) + {}^{x}C_2\, \Delta^2 f(0) + {}^{x}C_3\, \Delta^3 f(0) + \ldots + {}^{x}C_n\, \Delta^n f(0).$

$= \sum_{k=0}^{k=n} \binom{x}{k} \Delta^k f(0)$; = NF [n] (x); say. (2.1)

Wherein, 'Binomial Coefficients ${}^{x}C_k$ (${}^{x}C_n$)' represent a polynomial of degree k (n) in 'x', respectively. And "NF [n] (x)" is the "Newton's Forward Difference Interpolation Polynomial (of degree 'n' in 'x')".

Firstly, we will take of the 'Statistical Concept' of "Reduced-Bias", like in [5]/ Sahai (2004). They, too, proposed an "Iterative Algorithm" which happened to be using, a-posteriori, the aforementioned known information "namely, 'f (k/n)'; k = 0 (1) n", iteratively.

In fact, if at any stage/iteration the current polynomial interpolator "$O_n$ (f) (x)", is using the known values, say v (k) [= f (k/n); k =0 (1) n]'s; this algorithm will lead to the less-biased polynomial. Incidentally, at any stage/ iteration the 'Interpolation-Error' is partly due to the "Bias".

Analogously to [5]/ Sahai (2004), to illustrate in our application, let us consider the beginning-of-the-beginning, namely the very first "Iteration", as detailed in what follows.

Using our "Newton's Forward Difference Interpolation Polynomial" 'NF [n] (x)', we could get the interpolated values, say 'v (k/n)' for the unknown function 'f' at the 'n+1 equidistant-knots' "k/n [k = 0 (1) n]" in the 'Interpolation-Interval C [0, 1]'. Now, a-posteriori to the manufacture of our aforesaid "Newton's Interpolation Polynomial 'NF [n] (x)' ", using the available information, namely the given values 'f (k/n)' of the unknown function 'f' at the 'n+1 equidistant-knots' "k/n [k = 0 (1) n]"; we use this very information [so-to-say more fully] again to calibrate the "Error-Values" at those 'n+1 equidistant-knots' "k/n [k = 0 (1) n]", as below.

Say, Er (k/n) = v (k/n) – f (k/n); [k = 0 (1) n]. (2.2)

Now, using these "Error-Values Er (k/n); [k = 0 (1) n]", in (2.1), we can have the "Error-Interpolation Polynomial; Say Er [n] (x)", using (2.1), i.e. the same "Newton's Forward Difference Interpolation Polynomial" 'NF [n] (x)', again! Hence we could arrive at the "Reduced-Error/ Less-Biased Interpolation-Polynomial":

Say IP [(1-1)] (x) = NF [n] (x) - Er [n] (x). (2.3)

The achievement in (2.3) is 'Step # 1' in "Iteration # 1"/ Top-Layer 'Bread' of "First-Sandwich"! 'Step # 2' in "Iteration # 1"/ The 'Stuffing' for "First-Sandwich" comprises of the achievement per the application of the "Second Statistical Perspective of 'Reduced Relative Mean-Square-Error (RMSE)'", as detailed in what follows.

Analogously to the estimation strategy in [6]/ Searles (1964), we consider the scalar-multiplication-variant "b. IP [(1-1)] (x)" of the 'Interpolation-Polynomial' in (2.3).

Likewise, we will try to minimize/ reduce the "Summed-Squared-Relative-Error":

Say, SSRE (1) [n] (x) = $\sum_{k=0}^{k=n} [(Est(1-1)\left(\frac{k}{n}\right) - f\left(\frac{k}{n}\right))^2]/(f(\frac{k}{n}))^2$ = $A^*b^2 - B^*b + c$. (2.4)

As, we would have to rule out the choice of the possibility of imaginary/complex roots of the "Quadratic in 'b' " in (2.4); the following choice-value of 'b' will be reducing the 'RMSE'/ "Summed-Squared-Relative-Error" in (2.4).

Say, $b_0 = (B/ 2^*A)$. (2.5)

 

Thus, the 'Step # 2' in "Iteration # 1"/ the 'Stuffing' for the "First-Sandwich" happens to be the following "Interpolation Polynomial", using (2.5) in (2.3):

Say, IP [(1-2)] (x) = $b_0$ * IP [(1-1)] (x) (2.6)

Again, after this achievement per (2.6) using the "Statistical Perspective of Reduced RMSE", the 'Step # 3' in "Iteration # 1"/ Bottom-Layer 'Bread' of the "First-Sandwich" will consist in the use of the "Statistical Perspective of Reduced-Bias" again, like in the 'Step#1'! This would be exactly analogously to the 'Step # 1' in "Iteration # 1"/ Top-Layer 'Bread' of "First-Sandwich", with "IP [(1-2)] (x)" in the shoes of "NF [n] (x)", and Err IP [(1-2)] (x) in the shoes of Err NF [n] (x), analogously to (2.3)!

This leads us to "First-Sandwich Interpolation-Polynomial at Iteration # 1":

Say, NF [1, n] (x) ≡ IP [I#1] (x) = IP [(1-2)] (x) - Er IP [1-2] (x). (2.7)

Now, the improvement, achieved by an a-posteriori use of the information, as above in the three steps, happens to be happening iteratively.

For example, similarly to (2.7), the "Second-Sandwich Interpolation-Polynomial at Iteration # 2" [for convenience of notation, we will have the following equivalent notation for the relevant Interpolation-Polynomial]:

Say, NF [2, n] (x) ≡ IP [I#2] (x) = IP [(2-2)] (x) - Er IP [2-2] (x). (2.8)

And, at the beginning-of-the-beginning of the 'Iteration # 2', "IP [I#1] (x)" will be in the shoes of "NF [n] (x)", at the Step#1, and the subsequent Steps # 2 & 3 will follow exactly analogously.

Any typical iteration uses the twin statistical concepts of 'Reduced Relative Mean Square Error (RRMSE) and that of 'Reduced-Bias (RB); the former concept being sandwiched by the latter.

Thus, for convenience of notation, we will have the following equivalent for the "Typical-Sandwich Iterative Newton's Interpolation-Polynomial at Iteration # J":
Say, NF [J, n] (x) ≡ IP [I#J] (x) = IP [(J-2)] (x) - Er IP [J-2] (x); [J = 1 (1) m]. (2.9)

Wherein "m" will be the arbitrary "Number of Iterations" we are pleased to go for!

In the following section, the potential of the achievable improvements through the proposed 'Computerizable sandwich- iterative numerical algorithm for improved interpolation polynomial using Newton's forward difference formula' is tried to be brought out/ illustrated per an 'Simulation Empirical Study' for which the function 'f' is assumed to be known in the sense of 'Simulation'.

## 3. THE EMPIRICAL SIMULATION STUDY:

To illustrate the gain in efficiency of interpolation by using our proposed 'Computerizable sandwich- iterative numerical algorithm for improved interpolation polynomial using Newton's forward difference formula', we have carried an empirical simulation study. We have taken the example-cases of n = 2, 3, 5, and 8 (i.e. n + 1 = 3, 4, 6, and 9, knots) in the empirical study to numerically illustrate the relative gain in efficiency by using the Algorithm vis-à-vis the 'Original Newton's forward difference interpolation formula' in each example-case of the n-value.

*Essentially, the empirical study is a simulation one wherein we would have to assume that the function, being tried to be interpolated, namely f (x) is known to us.*

The numerical metric of the improvement is developed using the sum of the absolute {absolute differences between the actual (knowable, as the function is assumed to be known in the sense of the simulating nature of the empirical study) and the interpolated values at the mid-points of the equidistant-points/ knots in the 'Interpolation-Interval', say [0, 1]}.

Once again we have confined to illustrations of relative gain in efficiency by the Iterative Improvement for the following four illustrative functions: f (x) = exp(x), ln (5 + x), sin (5 +x), and $5^x$. *To illustrate the POTENTIAL of improvement with our proposed Sandwich-Iterative Algorithm, we have considered THREE Iterations only.*

 

The numerical values of SEVEN quantities are tabulated ~ THREE Percentage Relative Errors (PRE's) corresponding to our Improvement Iteration (I # = 1, or 2, or 3) (PRE_NF[I (#), n] (x)), to the Original Newton's Interpolation Polynomial (PRE_NF [n] (x)), and the corresponding THREE Percentage Relative Gains (PRG's), corresponding to our Improvement Iteration (I(#) = 1, or 2, or 3), by using our Sandwich-Iterative Algorithmic Interpolation Polynomial in place of Original Newton's Interpolation Polynomial (PRG_NF[I(#), n] (x); I(#) = 1(1)3).

For the calibration of these, the respective formulae are defined as in the following details. The 'PRE_NF[n]' of "Original Newton's Forward Difference Interpolation Formula":

$$\text{PRE\_NF}[n] = \left[\frac{\sum_{k=1}^{k=n} \ abs.\{(Et\ \text{NF}\,[\text{n}]\ f\,(\frac{2*k-1}{2*n}) - f\,(\frac{2*k-1}{2*n}))\}}{\{\sum_{k=1}^{k=n} \text{abs.}[\,f\,(\frac{2*k-1}{2*n})]\,\}}\right]*100\% \tag{3.1}$$

The 'PRE_ NF[I (#), n]' of the 'Interpolation Formula IF (I) N[n] f(x)' after using Improvement Iteration (I # 1, or 2, or 3) on Original Newton's Forward Difference Interpolation Formula using n intervals in [0, 1], i.e. [(k- 1)/n, k/n]; k= 1(1) n:

$$\text{PRE\_NF}\,[\text{I}\,(\#), \text{n}] = \left[\frac{\sum_{k=0}^{k=n} \ abs.\{(Et\ \text{NF[I(\#),n]}\ f\,(\frac{2*k-1}{2*n}) - f\,(\frac{2*k-1}{2*n}))\}}{\{\sum_{k=0}^{k=n} \text{abs.}[\,f\,(\frac{2*k-1}{2*n})]\,\}}\right]*100\%;\ \text{I}\,(\#) = 1/2/3 \tag{3.2}$$

And, hence, the Percentage Relative Gain (PRG), defined exactly analogously to PRE, by using the proposed Improvement Iteration [: I (#) (e.g., = 1, or 2, or 3)] Polynomials with the n intervals in [0, 1] over using the Original Newton's Forward Difference Interpolation Polynomial could be defined as below:

$$\text{PRG\_ NF}\,[\text{I}\,(\#), \text{n}] = \left[\frac{(\text{PRE\_NF[I (\#),n](x)} - \text{PRE\_NF[n](x)})}{(\text{PRE\_NF[n](x)})}\right]*100\%;\ \text{I}\,(\#) = 1/2/3 \tag{3.3}$$

These SEVEN values are computed using *MAPLE RELEASE 13*, for each example-number 'n' of the intervals in [0, 1].

These values are tabulated in the following four tables ~ Tables 1 to Table 4, respectively, for each of the four illustrative functions, namely, f (x) = exp(x), ln (5 + x), sin (5 +x), and $5^x$, in the APPENDIX.

## 4. CONCLUSION:

As per the four Tables in the APPENDIX, the PRE's for our Algorithmically-Improved Sandwich-Iterative Polynomial Interpolations are PROGRESSIVELY lower on each of the three successive iterations, as compared to that for the Original Newton's Forward Difference Interpolation Ploynomial, for all the illustrative functions. The PRG's due to the use of our proposed Algorithmic Sandwich-Iterative Polynomial Approximations vis-a-vis Newton's Forward Difference Interpolation Ploynomial are also, therefore, PROGRESSIVELY increasing on each of the three successive iterations, for all the illustrative functions. Other "Polynomial Interpolation Formulae" could also be improved, similarly!

## REFERENCES:

[1] N.L. Carothers, A Short Course on Approximation Theory, Bowling Green State University, Bowling Green, OH, 1998.

[2] N.L. Carothers, Real Analysis, Cambridge University Press, 2000.

[3] W. Cheney, D. Kincaid, Numerical Mathematics and Computing, Brooks/Cole Publishing Company, 1994.

[4] B.F. Polybon, Applied Numerical Analysis, PWS-KENT, 1992.

[5] A. Sahai, An iterative algorithm for improved approximation by Bernstein's operator using statistical perspective, Applied Mathematics and Computation 149 (2004) 327-335.

[6] D.T. Searles, The utilization of a known coefficient of variation in the estimation procedure, JASA, Vol. 59, 1964.

[7] A. Sheilds, Polynomial Approximation, The Math. Intell. 9 (3) (1987) 5–7.

**APPENDIX:**

**Table: 1**

(Iterative) Algorithmic (In %) Relative (Absolute) Efficiency/Gain for f (x) = exp(x).

| Items ↓ | n→ 2 | 3 | 5 | 8 |
|---|---|---|---|---|
| PRE_NF[n] | 24.44002684 | 30.61299450 | 35.47116298 | 37.96882561 |
| PRE_ NF[I (1), n] | 6.125213588 | 13.05557724 | 21.75581702 | 28.09593283 |
| PRE_ NF[I (2), n] | 1.675484265 | 5.501236536 | 11.90780026 | 18.50187196 |
| PRE_ NF[I (3), n] | 0.561457959 | 3.124961914 | 6.553647706 | 11.18295982 |
| PRG_ NF[I (1), n] | 74.93777879 | 57.35282532 | 38.66618630 | 26.00262879 |
| PRG_ NF[I (2), n] | 93.14450728 | 82.02973401 | 66.42963112 | 51.27088694 |
| PRG_ NF[I (3), n] | 97.70271136 | 89.79204106 | 81.52401230 | 70.54699575 |

**Table: 2**

(Iterative) Algorithmic (In %) Relative (Absolute) Efficiency/Gain for f (x) = ln (5+x).

| Items ↓ | n→ 2 | 3 | 5 | 8 |
|---|---|---|---|---|
| PRE_NF[n] | 2.690765699 | 3.607253111 | 4.358505803 | 4.787721971 |
| PRE_ NF[I (1), n] | 0.574736235 | 1.451305921 | 2.637263321 | 3.543861461 |
| PRE_ NF[I (2), n] | 0.098117145 | 0.539384043 | 1.545033545 | 2.577755072 |
| PRE_ NF[I (3), n] | 0.016605839 | 0.169801362 | 0.870327827 | 1.842831174 |
| PRG_ NF[I (1), n] | 78.64042064 | 59.76700619 | 39.49157257 | 25.98021601 |
| PRG_ NF[I (2), n] | 96.35356044 | 85.04723604 | 64.55130233 | 46.15904834 |
| PRG_ NF[I (3), n] | 99.38285823 | 95.29277939 | 80.03150928 | 61.50922745 |

**Table: 3**

(Iterative) Algorithmic (In %) Relative (Absolute) Efficiency/Gain for f (x) = sin (5+x).

| Items ↓ | n→ 2 | 3 | 5 | 8 |
|---|---|---|---|---|
| PRE_NF[n] | 24.11034758 | 31.70614414 | 36.40539866 | 38.67117409 |
| PRE_ NF[I (1), n] | 11.51168065 | 22.15319452 | 30.54648537 | 34.99845674 |
| PRE_ NF[I (2), n] | 6.290660795 | 16.78890100 | 27.01909918 | 32.68628893 |
| PRE_ NF[I (3), n] | 3.688348197 | 13.27135991 | 24.68735316 | 31.15018783 |
| PRG_ NF[I (1), n] | 52.25419040 | 30.12964799 | 16.09352872 | 9.497299827 |
| PRG_ NF[I (2), n] | 73.90887552 | 47.04843034 | 25.78271307 | 15.47634718 |
| PRG_ NF[I (3), n] | 84.70221888 | 58.14262418 | 32.18765878 | 19.44855939 |

**Table: 4**

(Iterative) Algorithmic (In %) Relative (Absolute) Efficiency/Gain for f (x) = $5^x$.

| Items ↓ | n→ 2 | 3 | 5 | 8 |
|---|---|---|---|---|
| PRE_NF[n] | 39.04602595 | 45.98918720 | 52.33547903 | 55.35706921 |
| PRE_ NF[I (1), n] | 8.156150712 | 17.40900204 | 30.10983275 | 39.43415362 |
| PRE_ NF[I (2), n] | 2.644073050 | 7.208612061 | 13.45360466 | 21.29285232 |
| PRE_ NF[I (3), n] | 1.429322649 | 5.430660794 | 8.353815462 | 11.50119439 |
| PRG_ NF[I (1), n] | 79.11144473 | 62.14544527 | 42.46764660 | 28.76401482 |
| PRG_ NF[I (2), n] | 93.22831713 | 84.32541974 | 74.29352915 | 61.53544141 |
| PRG_ NF[I (3), n] | 96.33939021 | 88.19143994 | 84.03794975 | 79.22362120 |

***********

 